



ALEPH VERSION 18.01

# How to Use the Patron Loader Interface - PLIF (file-20)

---

**Ex Libris**

© Ex Libris Ltd., 2007  
Last Update: February 4, 2008  
Document Version 1.4  
Code: A-ver18-HTPLIF-1.4

## CONFIDENTIAL INFORMATION

The information herein is the property of Ex Libris Ltd. or its affiliates and any misuse or abuse will result in economic loss. **DO NOT COPY UNLESS YOU HAVE BEEN GIVEN SPECIFIC WRITTEN AUTHORIZATION FROM EX LIBRIS LTD.**

This document is provided for limited and restricted purposes in accordance with a binding contract with Ex Libris Ltd. or an affiliate. The information herein includes trade secrets and is confidential.

## DISCLAIMER

The information in this document will be subject to periodic change and updating. Please confirm that you have the most current documentation. There are no warranties of any kind, express or implied, provided in this documentation, other than those expressly agreed upon in the applicable Ex Libris contract.

Any references in this document to non-Ex Libris Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this Ex Libris product and Ex Libris has no liability for materials on those Web sites.

Copyright Ex Libris Limited, 2007. All rights reserved.  
Documentation produced December 2007  
Document version 1.4  
Web address: <http://www.exlibrisgroup.com>

# Table of Contents

<b>OVERVIEW .....</b>	<b>4</b>
Determining the Match Point.....	4
Data Elements in the User Section .....	4
Data Elements in the Address Section.....	5
Data Elements in the Borrower Section.....	5
Assuring Uniqueness of Z308 IDs.....	6
<b>ACTIONS .....</b>	<b>7</b>
Using the Spaces and Ignore (Null) Characters.....	7
Post-PLIF .....	8
Sample Line .....	8
<b>RECORD STRUCTURE FOR THE PLIF INPUT FILE.....</b>	<b>8</b>
Notes (on Record Structure) .....	16
<b>RUNNING P_FILE_20 FROM THE COMMAND LINE .....</b>	<b>18</b>
<b>FROM THE PLIF (FILE-20) ONLINE HELP .....</b>	<b>19</b>
<b>WORKFLOW OF THE JOB .....</b>	<b>19</b>
<b>PERMISSIONS - ADDITIONAL INFORMATION .....</b>	<b>20</b>
<b>DELETING USERS - ADDITIONAL INFORMATION .....</b>	<b>20</b>
<b>INSERTING USERS - ADDITIONAL INFORMATION .....</b>	<b>21</b>
<b>BARCODES .....</b>	<b>21</b>
<b>ADDRESSES .....</b>	<b>21</b>
<b>MULTI-ADM ENVIRONMENT .....</b>	<b>21</b>
<b>PLIF ERROR CODES .....</b>	<b>21</b>

## Overview

Ex Libris provides a utility, `p_file_20`, to load patron records in machine-readable format into ALEPH. Most commonly, the records are student and faculty/staff records that come from the campus bursar's office on campus. The `p_file_20` job handles patron records which are in a pre-defined format, referred to as the **Patron Loader Interface Format (PLIF)**.

The system librarian should work with their Campus IT department to rearrange the bursar export data into the PLIF format. In almost all cases, there are some mappings that need to be determined by the system librarian based on ALEPH configuration tables.

## Determining the Match Point

The system librarian should consult the conversion specifications to find out what are the Patron ID (Z308) types. A typical arrangement of Z308 ID types might be:

- Type 00 – ALEPH, system assigned
- Type 01 – Patron's barcode number
- Type 02 – Campus ID number
- Type 03 – Social Security number

The IT department and the system librarian need to determine what number ALEPH and the bursar system have in common to provide a match point. Most likely the match point is a campus ID number. Type 00 is internal to ALEPH, and the bursar does not know this number. The barcode might change if the student replaces his ID card. Some campuses prohibit the use of a Social Security number to identify a person.

Alternatively, the system librarian may look in the online GUI client at the Additional IDs of a sample converted record to determine what numbers they have in common with the bursar.

Once the IT staff and the library have determined which ID they have in common with the ALEPH patron record, they use that as the match point to overlay a patron record in ALEPH with the incoming PLIF data.

Note that The MATCH-ID is used only for matching. The loader program does not add this ID automatically. If you want to be able to match on an ID later (for updating purposes), you need to add the ID using the ID section (described below). IDs of type 00 are created automatically, as they are the unique ALEPH system number.

## Data Elements in the User Section

The User section of the PLIF is mostly standard field translations (name, birth date) or default values (ILL borrowing limit). The library should give the default values for ILL borrowing, given that there might be different values for faculty and for students, for example, a 1000 limit for faculty but a student limit of 100.

**USER-PROFILE** The PROFILE is what determines default settings when a patron logs into the Web OPAC. Profiles are set in the ADMIN client. A typical example

might be that faculty members have a profile of FACULTY, which allows them to save course reserve files to the server. This is a privilege set in the ADMIN client for that profile.

**USER-DELINQ** The bursar can only send one delinquency (block) to ALEPH. The library might choose to do this if the student is in bad academic or financial standing and the library staff wants to know so they can limit his borrowing privileges. There are three delinquency fields in the ALEPH patron record, so if the bursar is sending one of them, the library staff needs to specify which of the fields to populate. If they want to populate the first delinquency field, then USER-REC-DELINQ-INDEX would be '1', the USER-REC-DELINQ would be the delinquency code (taken from xxx50/tab/tab\_delinq.eng), and the USER-REQ-DELINQ-NOTE would be the textual description of this block.

**USER-FIELD** and **USER-NOTE** fields are similar in structure to the USER-DELINQ fields.

**USER-REC-PROXY-FOR-ID** and **USER-REC-PRIMARY-ID** are used in the creating proxy borrowers. The ALEPH functionality behind this is that a student may be given privileges to check out materials for a faculty research professor without actually carrying the professor's card to the library, and without having the checked-out items appearing on the student's record.

## Data Elements in the Address Section

Address of type 01 is for permanent (parent's home) and type 02 is for campus (dorm/office) address. The address expiration dates should be checked with library staff, but it would be based on the student's registration, for example, if they are registered for fall term, the expiration date might be mid-January. Then, in the January loads, the date is again updated to mid-June if they registered for the Spring term. Faculty would probably have expiration dates far into the future.

## Data Elements in the Borrower Section

The Borrower record section also requires discussion between the IT department and the librarian. The IT department needs to translate patron statuses (Faculty, Staff, Undergrad, Grad, and so on) into the values in ALEPH. The patron values are listed in tab31, and the descriptions are in pc\_tab\_exp\_field\_extended.eng:

BOR-STATUS	L Undergrad Student	01
BOR-STATUS	L Grad Student	02
BOR-STATUS	L Faculty - Staff	03
BOR-STATUS	L Community	05
BOR-STATUS	L CMLE	19
BOR-STATUS	L ILL	20
BOR-STATUS	L Retired Fac - Staff	26

Faculty members probably have an ALEPH record of 03 and a XXX50 record of 03. Undergrads would have an ALEPH record of 01 and an XXX50 record of 03. Consortia may have different values for ALEPH and XXX50 records, and if the librarian is unable to determine the mapping, he should consult his Ex Libris project

librarian. Expiration dates in the Borrower section should be determined in a similar manner to address expiration dates.

Records that are updated or inserted take Borrower defaults from tab31, based on the Borrower Status and Sublibrary, with the exception of the expiration date.

## Assuring Uniqueness of Z308 IDs

The z308\_rec\_key (and z303\_rec\_key) must be unique. If it is not, you get an Oracle Unique constraint violated error.

p\_file\_20 matches on the USER-REC-MATCH-ID (usually the university ID/SSN). If a barcode (or other ID) in the PLIF record already exists for another user, p\_file\_20 detects this condition and issues an error message.

The Oracle z308 unique constraint error occurs when p\_file\_20 has generated an ID which it expects to be unique but is not.

There are two cases where p\_file\_20 generates an ID:

- It always generates a Z303 / Z308 type '00' ALEPH user number.
- If there is no ID type '01' (barcode) in the input file or if its ID-NO field is blank, p\_file\_20 generates a Z308 type '01' barcode.

The value for the ALEPH user number comes from the UTIL G/2 last-bor-id counter for the ADM library.

The value for the barcode (if not in the input record) comes from the UTIL G/2 last-bor-id-1 counter for the ADM library.

Thus, this error would occur when one of these counters is gives a value which duplicates an existing value.

Prior to version 16 or in a single-ADM version 16 installation, one can use this SQL to locate the highest value for the Z308:

```
SQL> select max (z308_rec_key) from z308 where substr (z308_rec_key,1,2) = '01';  
<'01' would be for the barcode; '00' for the ALEPH user ID>
```

### Note:

In version 16 and up, the Z308 is in the \$usr\_library

In the case of a multi-ADM installation in version 16 and up, the last-bor-id/last-bor-id-1 counters in the various ADM libraries need to be synchronized with each other to make sure that their values do not overlap. There are two ways to do this:

- Specify a different prefix for each ADM.
- Specify a different starting key number for each ADM which does not collide with the others.

## Actions

Each section above has an action associated with it. When an action is sent to the `p_file_20` job, the data in ALEPH is handled according to the associated section.

- A – Update if a match is found, and if a match is not found, add a new record with the same information. (Combination of I and U) This is the most common action.
- U – Update all fields in input record. If a record is sent with a U and there is no existing record, an error is reported.
- I – Insert the record. If the record exists already, the insert fails and the record not is updated.
- D – Delete the section of the record.
- X – Do nothing.

There is an action associated with every subsection of an input line. It is possible to start with a user record that has only a matching ID and action X and to follow that with address lines for update. This allows selective update of records associated with a patron.

## Using the Spaces and Ignore (Null) Characters

The PLIF format allows for two different characters:

- Space Character – Overwrites a data field in ALEPH with spaces
- Ignore (Null) Character – Ignores the data field; that is, leaves the data in the ALEPH record as is.

For example, the PLIF space character could be `%` and the ignore (Null) character could be `+`, which would be specified in the submission of the job `p_file_20`. Accordingly, use the single character `%` in the input file to indicate that the ALEPH field in the database should be blanked out (its value becomes spaces), no matter what is in it. Use the single character `+` in the PLIF input file to indicate that the ALEPH field in the database should be ignored.

The ignore character is useful if you know that librarians routinely update a field, like the user's e-mail address, when the data in ALEPH is assumed to be more accurate than the data from the bursar's office.

The spaces character and the ignore character cannot be any of the following: `* ? { } [ ] ( ) " ' ` ^ & > < , .` They have other meanings in UNIX. Either the spaces character or the ignore character can be the actual space character `_` if desired.

A clarification about how those two parameters work:

- The space and ignore (Null) characters operate at the whole field level, not at the individual character level. The field needs just a single occurrence of the space character. For example, if the patron previously had an invalid e-mail address and there was no replacement in the PLIF file, then you would put a single `%` in the email-address field. The patron's e-mail is blanked. (If the field has all `%%%%%%`'s, it may work, but we do not recommend this.)

- Similarly, if you have a blank or space character for the ignore (Null) character, `p_file_20` interprets a field which has blank as the first character and no other data, as an ignore character. For example, user-rec-name should appear in the input file as “Smith, John ” (up to 200 characters). If it begins with a blank (space) as in “\_Smith, John ” (Blank + text up to 200 characters) then `p_file_20` only takes the first character.

## Post-PLIF

No indexing jobs need to be run after execution of `p_file_20`. The only index involved (Z111 - User Record Keywords) is updated automatically.

## Sample Line

The following sample record uses the ‘A’ action to update someone by matching on ID type 03, but if that ID type 03 is not found, a new patron record is added. The whole line is structured as below with spaces and line breaks added for readability, but the whole record is in fact one line. Also, spaces have been compacted for readability.

User section:

```
A 03 12345          Swanson, Kristin          1000      1000      Y
010202  <--- 1 ID record follows (01), 2 address records follow (02),
and 2 borrower records follow (02)
```

ID section:

```
A 03 12345          AC  Y
```

Address section:

```
A 01 01 Kristin Swanson          123 Elm Lane          Little Falls, MN
A 02 02          Kristin Swanson          345 Crownhart Hall
Campus Dorm
```

Borrower Section (Kris Swanson is an undergrad registered for fall term so her expiration date is two weeks after semester ends)

```
A LIB50 01          20040115
A ALEPH 01          20040115
```

If a student drops out, you can just update the Borrower section so instead of expiring in January, it expires TODAY. Or if a faculty member quits, you would change the expiration date to TODAY as well.

## Record structure for the PLIF Input File

The PLIF input file may be in one of two formats:

- Sequential Input File
- The XML Format Input File

## Sequential Input File

The PLIF input file contains one line per one patron record. Fields are strung together with nothing between them, and there is an end of line character at the end of each record. The line is composed of four sections:

- **USER** (not repeatable) – Corresponds to the Z303 Oracle table. Contains information about the user that is only stored once – name, birth date, ILL library, and so on. At the end of the user record section, the loader specifies how many ID, ADDRESS, and BORROWER sections follow.
- **ID** (repeatable) – Corresponds to the Z308 Oracle table. There can be up to 99 types of User IDs.
- **ADDRESS** (repeatable) – Add one section for each address. Corresponds to the Z304 table.
- **BOR** (repeatable) – Determines a user's borrowing privileges. Corresponds to the Z305 table.

**Lengths** – USER: 1000 chars | ID: 100 chars | ADDR: 500 chars | BOR : 200 chars

The FILLER entries are to allow additional fields to be added without changing the structure of the input file.

**Padding of fields** – Type CHAR fields need to be right-padded with spaces (blanks); type NUM fields need to be left-padded with zeroes.

**Fields with pink-highlighting are new or changed in version 18.**

### USER SECTION: (Z303 – Global Patron Information) - 1000 chars

Field Name	Char / num	Length	Notes	Oracle
USER-REC-ACTION	CHAR	1	U, I, A, D or X (see Actions section on page 7)	(Mandatory)
USER-REC-MATCH-ID-TYPE	CHAR	2	01, 02, and so on. (See Note: on page 15.) See the previous explanation on "Determining the Match Point" (p. 4). Most commonly, this would be the SSN or university ID.	(Mandatory) Mandatory only for existing records (actions: A, U, D, X). Not mandatory for a new record, added by action I (or by action A if a match is not found)

Field Name	Char / num	Length	Notes	Oracle
USER-REC-MATCH-ID**	CHAR	20	From P. 15 notes: The MATCH-ID is used here <b>only</b> for matching. The ID is not added automatically here. If you want to be able to match on an ID later (for updating purposes), you need to add the ID using the ID section. IDs of type 00 is automatically created, because they are the unique ALEPH system number.	(Mandatory)
FILLER	CHAR	100	For future use	
USER-REC-NAME-TITLE	CHAR	10		Z303-TITLE
USER-REC-NAME	CHAR	200	For alphabetized list in the circ client, put in the form: Lastname, Firstname.	Z303-NAME (Mandatory)
USER-REC-BIRTH-DATE	NUM	8	YYYYMMDD format.	Z303-BIRTH-DATE
USER-REC-BUDGET	CHAR	20	Budget number for the patron. For information only.	Z303-BUDGET
USER-REC-EXPORT-CONSENT	CHAR	1	Set to "N" unless you are exporting records to other members of a consortium. ALEPH does not use; used only by user-written SQL.	Z303-EXPORT-CONSENT
USER-REC-DELINQ-INDEX*	NUM	1	(See Note: on page 15.)	
USER-REC-DELINQ*	NUM	2	(See Note: on page 15.)	Z303-DELINQ-N-1 Use zeros or relevant block number
USER-REC-DELINQ-N	CHAR	200		Z303-DELINQ-N-1
USER-REC-FIELD-INDEX*	NUM	1	(See Note: on page 15.)	
USER-REC-FIELD*	CHAR	200	(See Note: on page 15.)	
USER-REC-PROFILE	CHAR	12		Z303-PROFILE-ID

Field Name	Char / num	Length	Notes	Oracle
USER-REC-ILL-LIB	CHAR	5	Leave blank unless you are using the ILL module.	Z303-ILL-LIBRARY
USER-REC-HOME-LIB	CHAR	5		
USER-REC-ILL-TOTAL-LIMIT	NUM	4	Zeroes, unless you are using the ILL module.	Z303-ILL-TOTAL-LIMIT
USER-REC-ILL-ACTIVE-LIMIT	NUM	4	Zeroes, unless you are using the ILL module.	Z303-ILL-ACTIVE-LIMIT
USER-REC-SEND-ALL-LETT	CHAR	1	Y = all letters are sent to user. (Normal setting is "Y".) N= Hold, photocopy and ILL requests are not sent to user.	Z303-SEND-ALL-LETTERS
USER-REC-PROXY-FOR-ID	CHAR	12		Z303-PROXY-FOR-ID
USER-REC-PRIMARY-ID	CHAR	12		Z303-PRIMARY-ID
USER-REC-CON-LNG	CHAR	3	For North American sites normal setting is "ENG"	Z303-CON-LNG Default value: ENG
USER-REC-TYPE	CHAR	5	For future use Put blanks for now	Z303-USER-TYPE
USER-REC-PLAIN-HTML	CHAR	1	P=Plain; H=HTML; B=Both	Z303-PLAIN-HTML
USER-REC-WANT-SMS	CHAR	1	For future use Y=Yes; N=No  Put blanks for now	Z303-WANT-SMS
USER-REC-NOTE-INDEX	NUM	1	1 (update Z303-NOTE-1) Or: 2 (update Z303-NOTE-2)	
USER-REC-NOTE	CHAR	100	Free text note	Z303-NOTE-1 or Z303-NOTE-2 (according to the value set in USER-REC-NOTE-INDEX)
USER-REC-SALUTATION	CHAR	50	Text for mail, such as, "Dear ..."	Z303-SALUTATION
USER-REC-TITLE-REQ-LIMIT	NUM	4		Z303_TITLE_REQ_LIMIT
FILLER	CHAR	8		

Field Name	Char num	Length	Notes	Oracle
USER-REC-NO-ID	NUM	2	Number of ID sections which follow (Z308 - Patron's ID)	(Mandatory)
USER-REC-NO-ADDRESS	NUM	2	Number of Address sections which follow (Z304 - Patron's Address)	(Mandatory)
USER-REC-NO-BOR	NUM	2	Number of BOR sections which follow (Z305 - Patron's Local Information)	(Mandatory)

The last three fields of the USER section specify the number of additional sections following the user section on the same line (for example, 5 IDs, 3 addresses, and 1 borrower record).

**Note:**

If the number of additional sections is just a single digit, it needs to be preceded by a zero.

In the preceding case the last three fields would be "050301".

ID SECTION: (Z308 – Patron’s ID) - 100 chars

Field name	Char / num	Length	Notes	Oracle
ID-REC-ACTION	CHAR	1	U, I, A, D or X (see Actions section on page 7)	(Mandatory)
ID-REC-TYPE	CHAR	2	Use this type to identify what to match on later. Can be 01 (barcode), 02, 03 --> 99. You should <b>not</b> include a type 00 ID.	Z308-KEY-TYPE (Mandatory)
ID-REC-LOGIN	CHAR	20	This is the ID itself (barcode, SSN, University ID, and so on.). If this is blank, an ID is automatically created if tab10 line 39= yes. It should <b>not</b> be padded with zeroes; it should be left-aligned and right-padded with spaces. All letters need to be UPPERCASE.	Z308-KEY-DATA Z308_ID – part of Z308-KEY-DATA (Mandatory)
ID-REC-VERIFICATION	CHAR	20	A PIN code or other secondary ID; used primarily for Web access. Any letters need to be UPPERCASE.	Z308_VERIFICATION (Mandatory)
ID-VERIFICATION-TYPE	CHAR	2	This should always be “00” unless it is LDAP which is “02”	Z308-VERIFICATION-TYPE (Mandatory)
ID-STATUS	CHAR	2	AC = ACTIVE; NA = NOT ACTIVE	Z308_STATUS (Mandatory)
ID-ENCRYPTION	CHAR	1	Must be set to “Y” (for all new records)	Z308_ENCRYPTION (Mandatory)
FILLER	CHAR	52	For future use	

ADDRESS SECTION: (Z304 – Patron's Address) - 500 chars

Field name	Char / num	Length	Notes	Oracle
ADDR-REC-ACTION	CHAR	1	U, I, A, D or X (see Actions section on page 7)	(Mandatory)
ADDR-REC-SEQUENCE	NUM	2	Increment for each section added. (The first address section would be 01, the second, 02, and so on.)	(Mandatory)
ADDR-REC-TYPE	NUM	2	Type of address (By convention, the permanent address is 01 and the campus/mailling address is 02)	Z304-ADDRESS-TYPE (Mandatory)
ADDR-REC-ADDR-1	CHAR	50	First address line is name for notices – for example, John Smith	Z304-ADDRESS (Mandatory)
ADDR-REC-ADDR-2	CHAR	50		Z304-ADDRESS
ADDR-REC-ADDR-3	CHAR	50		Z304-ADDRESS
ADDR-REC-ADDR-4	CHAR	50		Z304-ADDRESS
ADDR-REC-ADDR-5	CHAR	50		Z304-ADDRESS
ADDR-REC-ZIP	CHAR	9		Z304-ZIP
FILLER	CHAR	1		
ADDR-REC-PHONE	CHAR	30		Z304-TELEPHONE
ADDR-REC-PHONE-2	CHAR	30		Z304-TELEPHONE-2
ADDR-REC-PHONE-3	CHAR	30		Z304-TELEPHONE-3
ADDR-REC-PHONE-4	CHAR	30		Z304-TELEPHONE-4
ADDR-REC-E-MAIL	CHAR	60		Z304-EMAIL-ADDRESS

Field name	Char / num	Length	Notes	Oracle
ADDR-REC-START-DATE	NUM	8	YYYYMMDD format. When the Z304 address record created from this section becomes active  <b>Note::</b> xxx50/tab/tab/bor/address is an alternative way to control start and stop dates.	Z304-DATE-FROM (Mandatory)
ADDR-REC-STOP-DATE	NUM	8	YYYYMMDD format. When the Z304 address record created from this section expires. (Note that the BOR-REC-EXPIRY-DATE controls when the borrower record expires.)	Z304-DATE-TO (Mandatory)
ADDR-REC-SMS-NUMBER	CHAR	30	For future use Phone no. to which a SMS message can be sent. Put blanks for now.	
FILLER	CHAR	9	For future use	

**BOR SECTION: (Z305 – Patron’s Local Information) - 200 chars**

Field name	Char / num	Length	Notes	Oracle
BOR-REC-ACTION	CHAR	1	U, I, A, D or X (see Actions section on page 7)	(Mandatory)
BOR-REC-SUB-LIBRARY	CHAR	5	For North American libraries, usually the xxx50 ADM library code (but could also be “ALEPH” or a valid sublibrary code)	Z305-SUB-LIBRARY (Mandatory)
BOR-REC-TYPE	CHAR	2	the z305_bor_type	Z305-BOR-TYPE
BOR-REC-STATUS	CHAR	2	the z305_bor_status	Z305-BOR-STATUS (Mandatory)
BOR-REC-EXPIRY-DATE	NUM	8	YYYYMMDD format. If blank, takes tab31 value.	Z305-EXPIRY-DATE (Mandatory)
BOR-REC-REGISTRATION-DATE	NUM	8	The date on which the patron registered in the library	Z305-OPEN-DATE
FILLER	CHAR	174	For future use	

## Notes (on Record Structure)

\* = There are 3 note fields in the Z303 table:

02 Z303-FIELD-1	PICTURE X(100).
02 Z303-FIELD-2	PICTURE X(100).
02 Z303-FIELD-3	PICTURE X(100).

If you specify "1" for the USER-REC-FIELD-INDEX, Z303-FIELD-1 is updated.

If you specify "2", Z303-FIELD-2 is updated.

If you specify "3", Z303-FIELD-3 is updated.

(The same principle is used for the Delinquency field.)

\*\* = The MATCH-ID is used **only** for matching. The loader program does not add this ID automatically. If you want to be able to match on an ID later (for updating purposes), you need to add the ID using the ID section. IDs of type 00 are automatically created because they are the unique ALEPH system number.

Records that are updated or inserted takes Borrower defaults from Tab31, based on the Borrower Status.

The Gender and Place of Birth fields may be inserted and updated by the Patron Loader (file-20) service only when using the XML input file. The fields cannot be modified in the Patron Loader when using a sequential input file.

## The XML Format Input File

The patron load interface file can be an XML format file. If XML input is used, the following format should be used:

- The file must include an opening root tag of <p-file-20> and a closing tag of </p-file-20>.
- Within the opening and closing tags, several patron records may be included, each one with an opening tag of <patron-record> and a closing tag of </patron-record>.
- Within each <patron-record> section, z303/z304/z305/z308 records can be included under the opening tags <z303>, <z304>, <z305>, <z308> and the closing tags </z303>, </z304>, </z305> and </z308> respectively.
- The <patron-record> section must include one <z303> section and any number of <z304>, <z305> and <z308> sections.
- Each <znnn> element must include a <record-action> element, with I/D/U/A as permitted values. These values have the same interpretation as explained above in the description of the flat file format.
- The z303 element in each patron record must include the tags <match-id> and <match-id-type>. These tags have the same interpretation as explained above in the description of the flat file format.
- The rest of the tags under each <znnn> element correspond to the available fields of each znnn.

Below is an example of an XML input file:

```
<?xml version="1.0"?>
<p-file-20>
  <patron-record>
    <z303>
      <match-id-type>00</match-id-type>
      <match-id>00000036</match-id>
      <record-action>A</record-action>
      <z303-id>00000036</z303-id>
      <z303-user-type>REG</z303-user-type>
      <z303-con-lng>ENG</z303-con-lng>
      <z303-name>Haugh, Sheldon</z303-name>
      <z303-title></z303-title>
      <z303-delinq-1>00</z303-delinq-1>
      <z303-delinq-n-1></z303-delinq-n-1>
      <z303-delinq-3>00</z303-delinq-3>
      <z303-delinq-n-3></z303-delinq-n-3>
      <z303-budget>BUDGET1</z303-budget>
      <z303-profile-id>PROF1</z303-profile-id>
      <z303-ill-library>U70WD</z303-ill-library>
      <z303-home-library>WID</z303-home-library>
      <z303-note-1>This patron has many overdue items</z303-note-1>
      <z303-ill-total-limit>0010</z303-ill-total-limit>
      <z303-ill-active-limit>0004</z303-ill-active-limit>
      <z303-birth-date>19470417</z303-birth-date>
      <z303-export-consent>Y</z303-export-consent>
      <z303-proxy-id-type>00</z303-proxy-id-type>
      <z303-send-all-letters>Y</z303-send-all-letters>
      <z303-plain-html>H</z303-plain-html>
      <z303-want-sms>N</z303-want-sms>
      <z303-title-req-limit>0010</z303-title-req-limit>
      <z303-gender>M</z303-gender>
      <z303-birthplace>United States</z303-birthplace>
    </z303>
    <z304>
      <record-action>A</record-action>
      <z304-id>00000036</z304-id>
      <z304-sequence>01</z304-sequence>
      <z304-address-0>Sheldon Haugh</z304-address-0>
      <z304-address-1>Harvard University</z304-address-1>
      <z304-address-2>Psychology Department</z304-address-2>
      <z304-address-3>William James Hall 1540</z304-address-3>
      <z304-address-4>Cambridge MA 02138 UNIV</z304-address-4>
      <z304-zip>02138</z304-zip>
      <z304-email-address>email@email_server.com</z304-email-address>
      <z304-telephone>38108831</z304-telephone>
      <z304-date-from>20010701</z304-date-from>
      <z304-date-to>20051031</z304-date-to>
      <z304-address-type>02</z304-address-type>
      <z304-telephone-2>6174953872</z304-telephone-2>
    </z304>
  </patron-record>
</p-file-20>
```

```

</z304>
<z305>
  <record-action>A</record-action>
  <z305-id>00000036</z305-id>
  <z305-sub-library>ALEPH</z305-sub-library>
  <z305-bor-type>AS</z305-bor-type>
  <z305-bor-status>01</z305-bor-status>
  <z305-registration-date>20050101</z305-registration-date>
  <z305-expiry-date>20060101</z305-expiry-date>
</z305>
<z308>
  <record-action>A</record-action>
  <z308-key-type>00</z308-key-type>
  <z308-key-data>00000036</z308-key-data>
  <z308-verification>00000036</z308-verification>
  <z308-verification-type>00</z308-verification-type>
  <z308-status>AC</z308-status>
  <z308-encryption>H</z308-encryption>
</z308>
<z308>
  <record-action>A</record-action>
  <z308-key-type>01</z308-key-type>
  <z308-key-data>B-034916</z308-key-data>
  <z308-verification>B-034916</z308-verification>
  <z308-verification-type>00</z308-verification-type>
  <z308-status>AC</z308-status>
  <z308-encryption>H</z308-encryption>
</z308>
</patron-record>
</p-file-20>

```

Any error in the XML format (like missing mandatory elements) causes the whole record to be ignored. Missing optional fields under elements are considered spaces.

The Gender and Place of Birth fields may be inserted and updated by the Patron Loader (file-20) service only when using the XML input file. The fields can be modified in the <z303-gender> and <z303-birthplace> tags in the <z303> section. The fields cannot be modified in the Patron Loader when using a sequential input file.

## Running p\_file\_20 from the Command Line

For the purpose of these instructions, assume that the input PLIF file is called `patrons.plif`.

### To run p\_file\_20 from the Command Line:

1. Copy the PLIF file to the `$data_scratch` directory of the ADM library whose patrons you want to update (for example, USM50). The name of the file must be in lowercase. For example:

```
dlib USM50
```

```
ds
cp /wherever/patrons.plif .
```

2. Enter the following command to run `p_file_20`:

```
csh -f $aleph_proc/p_file_20
USM50,patrons.plif,patron_update.rpt,Y,%,+ > & patron_update.log
&
```

The job takes seven parameters, separated by commas.

- 1 – Library in which to do patron update (for example, USM50)
- 2 – Name of PLIF file, which should be in the `$data_scratch` directory (for example, `patrons.plif`)
- 3 – Report file for update, which is left in the `$data_scratch` directory (for example, `patron_update.rpt`)
- 4 – Y to update the database, N to not change the database and only produce a report
- 5 – Spaces character (overwrite-with-blanks character). If this character is in the input file, a space is written to the ALEPH field
- 6 – Ignore character (null character). If this character is in the input file, the ALEPH field is protected.
- 7 – In version 16 and up there is an optional seventh parameter, character conversion routine.
- 8 – Input file type: XML or FLAT (Aleph sequential).

## From the PLIF (file-20) Online Help

ALEPH requires that each patron Z303 record has a matching Z308 record with a type '00' (internal system ID) and a type '01' (barcode). Therefore, when patrons are loaded using this service:

- If there is no Z308 type '00' reported in the input file, or already present in the record, the system generates a Z308 type '00' with KEY and VERIFICATION values the same as the Z303 ID. If only the verification is missing, it is generated with the same value as the Z308 KEY.
- If there is no Z308 type '01' reported in the input file, or already present in the record, the system generates a Z308 type '01' with KEY and VERIFICATION values the same as the Z308 type '00'. If only the verification is missing, it is generated with the same value as the Z308 KEY.

## Workflow of the Job

For each line of the input file (which contains all the relevant records for a specific patron):

1. Read all sections into ALEPH *Znnm* tables.

2. Find the Patron Internal ID, using the matching data given in the input.
3. Update/Insert Mode
4. In cases of Update mode, merge existing ALEPH *Znnn* table data with the input data (without updating the database yet).
5. In cases of Insert mode, assign default values to the incoming *Znnn* record from ALEPH configuration tables.
6. Check the validity of the desired actions:
7. Check each *Znnn* for record validity.
8. Check if the action flags set are consistent. For example, deleting a global-user together with inserting address, or updating a non-existent record are not valid sets of action flags.
9. Check permissions on actions (using tab100 permission flags and Z303-PLIF-MODIFICATION flags).
10. If no error is detected, update the database with the input data. In case of error, reject the input line and update the report-file.

## Permissions - Additional Information

For permission checks the following variables in tab100 are considered:

USER-ADDR-PERMISSION

USER-PERMISSION

These permission variables can be used to limit the user permission for updating the patron's address and global records by limiting permission to the ADM library of the patron's home library. This requires that the home library be registered in the global record. If either of the permission variables is inconsistent with the PLIF action, the input-line with all its records is rejected.

In addition, Z303-PLIF-MODIFICATION is used for blocking undesired modification. This block does not cause the input-line to be rejected; it simply causes it to be ignored for update purposes.

## Deleting Users - Additional Information

When deleting a patron, there is no need to specify all of the *Znnn* records to be deleted. Only the Z303 (the global-user information record) should be included. In other words, the input-line for the patron to be deleted should include only the matching data and the global user record (required only because it has the D flag in it). If other records are included in the input file they should appear with the action flag X or D; otherwise, they cause the input-line to be rejected.

Special checks are applied when deleting a patron, as follows:

*get\_balance* – A patron cannot be deleted when the patron's cash transactions are not balanced (i.e. they are not equal to zero, meaning that there must not be a debt to the library or vice versa).

`get_transferred_balance` – A patron cannot be deleted when the patron's transferred cash transactions are not balanced (meaning, they have a debt to the library or vice versa, waiting to be handled by an accounts receivable agency).

`get_buf_z36` – A patron cannot be deleted if there are loan records.

## Inserting Users - Additional Information

When a new record does not have the Z308 records that are required for a new patron (i.e., the "00" type, which is the login-by-internal-id, and the "01" type, which is the login-by-barcode), tab100 DEFAULT-BOR-ID is consulted. If the value is "N", the "00" type record must be supplied in the input-file. If the value is "Y", the system automatically generates the necessary Z308 records for the new patron. Note that the matching data is not considered login information, since it is lacking a verification field.

## Barcodes

Each patron can create a hold on one barcode which is z308-key-type 01 (or login-rec-type 01). The system should not allow insertion of two barcodes, or deletion of existing barcodes, since ALEPH assumes each patron has one and only one barcode. To change an existing barcode, the input file must contain the new barcode in Insert mode. `P_file_20` should recognize this action as "switch old barcode with the input-file barcode". In this case, if the input file does not contain any verification, the old verification (from the old barcode record) is copied to the new record and the old record is deleted completely. This deletion is shown in the report as well.

## Addresses

When updating an address, the input-file must contain the address-type. The system, assuming the input-file address contains an active address, should merge the input-record with the patron active address of the same address-type. In other words, the existing z304 record is overwritten only if it is active and the input-record is active as well. Otherwise, the system considers the case as a "no-match", meaning it produces an error if the action is an update or creates a new record if the action is an append.

## Multi-ADM Environment

When using the `delete_bor_total` routine in a multi-ADM environment, the patron records are deleted, but the global Z303 record remains in the database if the current library is shared and there are local (Z305) records in the other shared libraries.

## PLIF Error Codes

The following table lists the PLIF error codes and their interpretation:

Error Code	Error Text	Description
5000	Failure occurred while trying to read XML input file (XML might be invalid)	The input XML file does not match the expected structure.
5001	Successed to \$2 table \$1. cur-id \$3.	Processing the input file record is successful.
5002	io_\$1 routine failed while trying to \$2. cur-id \$3.	An Oracle IO problem has been detected while trying to write\update a record.
5003	Unable to process record due to problems in other records.	One of the uploaded records has an error, which caused the system to not attempt uploading of the other records. A separate error line reports the exact error and the record in which it is found.
5004	Record is ready to process. Update DB flag is off.	Uploaded record is valid but the database was not updated because the batch job's 'Update Database' flag is set to No.
5005	Patron id \$3 was successfully deleted. Z303 remains for shared patrons.	A shared patron was fully removed from the library in which the batch job has run. The patron's shared information remained undeleted because the patron is active in other shared libraries.
5006	delete_bor_total failed while trying to delete patron id \$3.	Full patron deletion failed. This can be either because the patron information was not found in the database or due to an Oracle error that happened when attempting the deletion.
5007	Patron id \$3 was successfully deleted.	Full patron deletion succeeded. Patron is not active in other libraries.
5010	Can not upd/del \$1. No match found \$3.	An update or deletion of patron information (Z303/Z304/Z305) was requested, but the match ID in the input file could not be matched against an existing ID in the database. This is similar to error 5016.
5011	Can not insert \$1. Match found \$3.	An Insert action was requested for some patron information, but the input ID was found to be an already existing ID. The record was rejected because Insert actions are valid only if the input ID is not an already existing one.  You should use the Update or Add action if an update is really intended.

Error Code	Error Text	Description
5012	Unsupported action "\$2".	<p>The input record included a non supported action instruction.</p> <p>The allowed actions are:</p> <ul style="list-style-type: none"> <li>• U-Update all fields in input record. If a record contains a U and there is no existing record, an error is reported.</li> <li>• I-Insert the record. If the record already exists, the insert will fail and the record is not be updated.</li> <li>• A-Add if no match is found, update if a match is found (Combination of I and U).</li> <li>• D-Delete the section of the record.</li> <li>• X-Do nothing.</li> </ul>
5013	Unable to delete due to cash.	Patron deletion failed because of unpaid cash charges.
5014	Unable to delete due to active loans.	Patron deletion failed because of existing loans.
5015	Failed to generate new user ID. Check tab100 or z52.last-bor-id.	<p>A new patron creation failed. The new patron record input did not include a patron ID. The system could not generate a new ID because of one of the following:</p> <ol style="list-style-type: none"> <li>1. The tab100 variable DEFAULT-BOR-ID in the uploading library is set to N. dsds</li> <li>2. The last-bor-id Z52 counter is not defined. It must be defined either in the ADM library or in the usr_library.</li> </ol>
5016	Cannot update/delete record. Login does not exist.	An update or deletion of patron login information (Z308) was requested, but the match ID in the input file could not be matched against an existing ID in the database. This is similar to error 5010.
5018	System cannot generate login for login-type other than 01.	<p>The input file includes a login record that is not a barcode (Z308 with Z308-KEY-TYPE is not 01) but has no login text. The record is rejected because the system cannot generate a login text that is not a barcode.</p> <p>If the uploaded login record is not a barcode then the input file must include the login text.</p>
5019	Unsupported z table "\$1".	The uploaded input file includes a record other than those allowed: Z303, Z304, Z305, or Z308.
5020	Unable to delete due to active holds	Local Patron Information (Z305) deletion failed because of existing hold requests in the deleted sublibrary local patron record.

<b>Error Code</b>	<b>Error Text</b>	<b>Description</b>
5021	check_\$1 routine failed. Can not ins/upd record. cur-id \$3.	<p>The input file patron information was merged with information in the database record, resulting in a non-valid record.</p> <p>This may happen with any one of the Z303, Z304, Z305, or Z308 records.</p> <p>Note the PLIF log file includes information about which record is non-valid and which one of its fields has the illegal value.</p>
5022	check_z308_duplicate routine failed. Cannot ins/upd record.	<p>The input file has one of the following errors in the login (Z308) record:</p> <ol style="list-style-type: none"> <li>1. The login text is already used by another patron. This is not allowed if the login type is defined in tab_bor_id (column 5) as usable for retrieving patrons in the GUI.</li> <li>2. The same login text AND password is already used by another patron. This is never allowed.</li> </ol>

Error Code	Error Text	Description
5023	System could not generate a "dummy" barcode. Check tab100 or z52.last-bor-id-1.	<p>This error may be caused by one of the following conditions:</p> <ol style="list-style-type: none"> <li>1. <ol style="list-style-type: none"> <li>a. The input file has a barcode record which does not include a login text AND</li> <li>b. The tab100 variable DEFAULT-BOR-ID-1 is defined in the ADM library as Y AND</li> <li>c. The last-bor-id-1 Z52 counter is not defined in both the ADM library and the usr_library</li> </ol> </li> <li>2. <ol style="list-style-type: none"> <li>a. The input file has a barcode record which does not include a password text AND</li> <li>b. The tab100 variable DEFAULT-BOR-VERIFY-2 is defined in the ADM library as Y AND</li> <li>c. The last-bor-verify-2 Z52 counter is not defined in both the ADM library and the usr_library</li> </ol> </li> <li>3. <ol style="list-style-type: none"> <li>a. The input file has a system ID record which does not include a password text AND</li> <li>b. The tab100 variable DEFAULT-BOR-VERIFY-1 is defined in the ADM library as Y AND</li> <li>c. The last-bor-verify-1 Z52 counter is not defined in both the ADM library and the usr_library</li> </ol> </li> </ol>
5024	Number of records specified in the input file is not numeric.	<p>The sequential input file format is used, but one of the following fields has invalid information:</p> <ul style="list-style-type: none"> <li>• USER-REC-NO-ADDRESS</li> <li>• USER-REC-NO-BOR</li> <li>• USER-REC-NO-ID</li> </ul>
5025	Update 'Share Users File' successful.	Relevant only to SUF users.
5026	Unable to update 'Share Users File'.	Relevant only to SUF users.

<b>Error Code</b>	<b>Error Text</b>	<b>Description</b>
5029	Cannot update/insert record when user record is being deleted.	The input file has a patron record with contradicting actions. The same patron record includes both a delete action for the global patron information and an update action for one of the patron details, such as the login or address. This is similar to error 5034.
5030	tab100 does not allow record updating (USER-PERMISSION or USER-HOME-PERMISSION)	Updating the patron records is not allowed because of one of the following: <ol style="list-style-type: none"> <li>1. The tab100 USER-HOME-PERMISSION variable requires that the patron belong to the updating library according to his Z303-HOME-LIBRARY, yet the updated patron does not belong to the updating library (ADM library).</li> <li>2. the tab100 USER-PERMISSION variable requires that the updated patron have a Local Patron Information record (Z305) in the updating library, but none exists.</li> </ol>
5031	tab100 does not allow record updating (USER-ADDR-PERMISSION or USER-HOME-PERMISSION)	Updating the patron address information is not allowed because of one of the following: <ol style="list-style-type: none"> <li>1. The tab100 USER-HOME-PERMISSION variable requires that the patron belong to the updating library according to his Z303-HOME-LIBRARY, yet the updated patron does not belong to the updating library (ADM library).</li> <li>2. The tab100 USER-ADDR-PERMISSION variable requires that the updated patron have a Local Patron Information record (Z305) in the updating library, but none exists.</li> </ol>
5032	SPACE character and NULL character cannot be the same.	The same value is used for the SPACE character and the NULL character.
5033	Cannot assign new barcode to an existing patron.	The input file includes an Insert action, attempting to add a new barcode to a patron that already has a barcode. A patron cannot have more than one barcode.
5034	Cannot update record when new user is being inserted.	The input file has a patron record with contradicting actions. The same patron record includes both an Insert action for the global patron information and an update action for one of the patron details, such as the login or address. This is similar to error 5029.

Error Code	Error Text	Description
5035	Cannot delete barcode from existing/new user.	The input file attempts to delete the barcode record. A patron must have a barcode record.
5036	Cannot insert more than one barcode.	The input file has more than one barcode record. A patron cannot have more than one barcode.
5037	Cannot update/delete record. Login record belongs to other patron.	The input file includes a Delete or Update action to a non existing login (Z308) record.
5040	tab100 does not allow record updating (USER-IDS-PERMISSION or USER-HOME-PERMISSION)	Updating the patron login information is not allowed because of one of the following: <ol style="list-style-type: none"> <li>1. The tab100 USER-HOME-PERMISSION variable requires that the patron belong to the updating library according to his Z303-HOME-LIBRARY, yet the updated patron does not belong to the updating library (ADM library).</li> <li>2. The tab100 USER-IDS-PERMISSION variable requires that the updated patron have a Local Patron Information record (Z305) in the updating library, but none exists.</li> </ol>
5041	Unable to delete due to ILL Request.	Patron deletion failed because of existing ILL requests.
5042	Successed to \$2 table \$1 with modification restrictions. cur-id \$3.	Processing the input file record is successful. Some of the input record's information was not updated. This is because some of the fields in the patron record are protected from PLIF update. This protection is done by setting the patron z303_plif_modification field.
5044	Cannot ignore user-record when there's no match id	An Ignore action was requested, but the match ID in the input file could not be matched against an existing ID in the database. This is similar to error 5010, but is triggered if the requested action is ignore.